

Research note 30

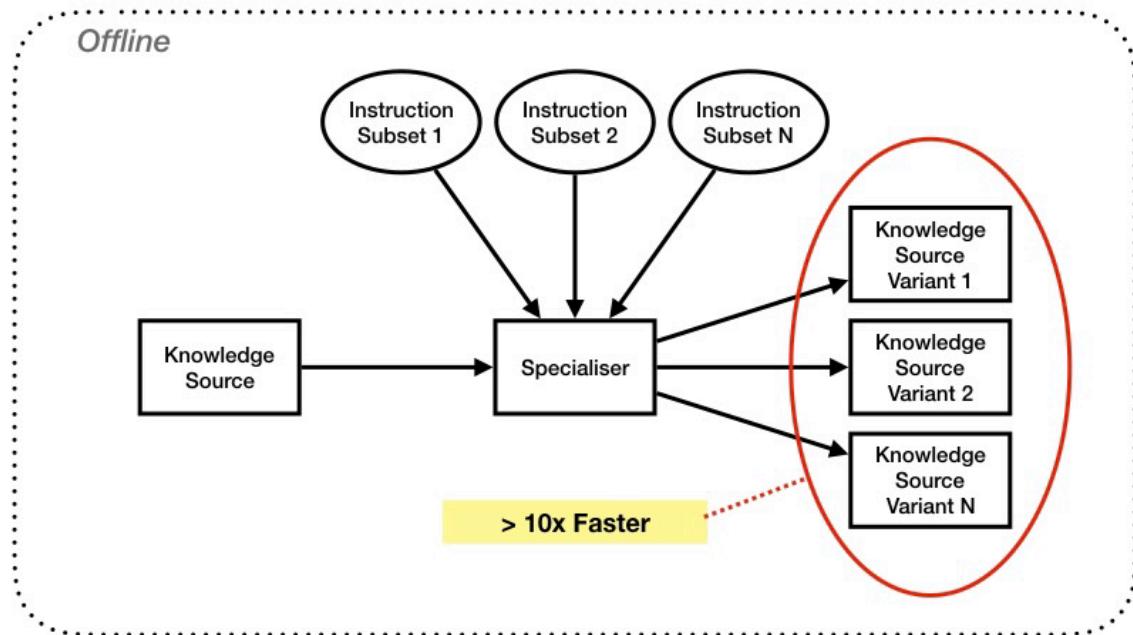
Partial Evaluation of Knowledge Sources

Edward McDaid & Sarah McDaid

12 Jan 2024

Zoea has used a program optimisation technique called partial evaluation to produce many additional variants of its knowledge sources - specialised for different instruction subsets. This improves Zoea runtime performance by over an order of magnitude.

Partial Evaluation of Knowledge Sources with Instruction Subsets



© zoea.co.uk

Zoea is an AI that generates code directly from a set of test cases. This is accomplished using a blackboard architecture that integrates a large number of autonomous experts (or knowledge sources). Each expert encodes a different aspect of software development or programming language knowledge. Experts are triggered by blackboard activations which

identify a knowledge source, a synthetic test case and an instruction set. The ability to specify a custom instruction set for every problem has existed in Zoa since the outset. Thus, knowledge sources can utilise the complete instruction set or any subset.

The biggest challenge facing Zoa and similar systems is the combinatorial explosion. There are a vast number of possible programs and this grows exponentially as program size increases. In response, Zoa has developed a unique technology that can predict the required subset of instructions for almost any problem, and do so in a relatively small number of attempts. This can reduce the number of candidate solutions that need to be considered by over 40 orders of magnitude. Informally, we call this approach 'instruction subsets'.

The ability to specify a custom instruction set for each knowledge source made the implementation the instruction subset approach in Zoa a trivial exercise. All we had to do was to generate many activations for each knowledge source, where each one has a different subset of instructions. This is how the first iteration of instruction subsets was implemented and it works very nicely. Instruction subsets dramatically reduce the size of the search space but the knowledge sources themselves still operate at the same speed.

At the same time, it is worth noting that the ability to specify an instruction subset for each knowledge source was originally put in place to support situations where the instruction set is formulated at runtime. In such cases the instruction subset that we will use is unknown until shortly before the knowledge source is activated. However, with the instruction subset approach - as it is currently formulated - we know well in advance exactly what subsets we will be using. This provides the opportunity for a significant performance improvement - by specialising the knowledge sources.

Different knowledge sources use custom instruction sets in different ways. However, in general terms, there is often some code and data relating to different instructions together with additional logic to handle their inclusion or exclusion. For example, some knowledge sources include the equivalent of a large switch statement that acts as a dispatcher for all of the instructions. This is a relatively expensive operation that occurs in a very hot code path - i.e. one that is executed a very large number of times. If the set of instructions for a given activation is already known, then much of this work is effectively redundant.

Partial evaluation is a program optimisation technique that transforms a piece of code at compile time based on pre-knowledge of some of the input values. In effect, the program is partially executed in advance - to the extent that the known input values allow. For example, some of the variables in the code can be replaced by any corresponding known values. This in turn can ossify the results of conditional expressions, possibly resulting in the complete removal of some code blocks, and so on. What we end up with is a different - often smaller - version of the input program that has been 'specialised' for the particular values of the input variables. Frequently, this specialised program is more efficient than the original code.

At the end of the day a Zoa knowledge source is a piece of code that takes a synthetic test case and an instruction set as inputs. We can therefore specialise any knowledge source with respect to a particular instruction subset. Depending on the subset size there can be hundreds or thousands of instruction subsets in a given family. This means that we will produce the same number of specialised knowledge source variants - one for each instruction subset. At runtime, the same number of activations is required - except that each one now targets a different knowledge source variant, rather than the same knowledge source with different instruction subsets.

Zoa knowledge sources are fairly small and have a well defined structure. This means that a very simple specialiser is sufficient to partially evaluate knowledge sources with respect to instruction subsets. The resulting code represents the second iteration of the implementation of instruction subsets in Zoa.

The Zoa codebase now includes a large number of automatically generated knowledge source variants - comprising over 5 million additional lines of code. More importantly, partial evaluation has improved knowledge source runtime performance by over an order of magnitude.

Learn more at zoea.co.uk